

ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE VISUALIZACIÓN DE INFORMACIÓN VIAL EN TIEMPO REAL, PARA EL OBSERVATORIO DE MOVILIDAD VIAL DE LA CIUDAD DE PEREIRA

Título en Inglés, Analysis, design and implementation of a real time graphical user interface of traffic information, for the “Observatorio de Movilidad Vial” of Pereira

RESUMEN

El Grupo de Investigación Sirius junto con el apoyo de Colciencias, se creó el proyecto Observatorio de Movilidad Vial que busca mediante la medición de variables del tráfico, establecer un modelo del sistema vial de la ciudad tanto espacial como temporalmente, para mejorar las políticas de gestión y administración de la movilidad de la ciudad.

Este proyecto busca satisfacer los requerimientos de software del sistema de visualización del Observatorio de Movilidad Vial, estableciendo como base un sistema de información geográfico, que soportará espacialmente y en tiempo real, cada uno de los elementos viales de la ciudad, para después visualizarlos de manera óptima en una interfaz de usuario.

PALABRAS CLAVES: ArcObjects, Históricos, Información Geográfica, Java, Malla Vial, Movilidad, NetBeans, Oracle, SIG, Tiempo Real, Tráfico.

ABSTRACT

Sirius Research Group with the support of Colciencias, created the project “Observatorio de Movilidad Vial”, looking through the measurement of traffic variables, establish a model of the city road system both spatially and temporally, to improve management policies and management of mobility in the city.

This project seeks to meet the requirements of software visualization system of “Observatorio de Movilidad Vial”, establishing the basis of a geographical information system that will support in real-time and spatially, each one of the roads elements of the city, and then display them optimally in a user interface.

KEYWORDS: ArcObjects, Geographic Information, GIS, Historical, Java, Mobility, NetBeans, Oracle, Real Time, Road Grid, Traffic.

1. INTRODUCCIÓN

El sistema vial de la ciudad de Pereira y todos los elementos que lo componen (calles, carreras, avenidas, semáforos, cebras, etc.), tienen en su parte urbana y rural un conjunto de reglas dirigidas hacia la permanencia y gestión de su organización, fluidez y seguridad. El personal encargado de garantizar el óptimo funcionamiento del sistema vial, necesita conocer para tal fin la información que el sistema transmite segundo a segundo a través de sus elementos.

La necesidad de hacer una correcta gestión de sistema vial, ha hecho que las administraciones municipales de las ciudades con alto flujo vehicular, creen organismos independientes para atender las problemáticas del tránsito y su movilidad. El factor incidente que hace de la movilidad un problema importante, es el hecho que la ciudad es un ser viviente cambiante y en constante

crecimiento, por lo que su modelamiento y control entra en el campo de lo probabilístico.

Por lo anterior, este proyecto pretende hacer de Pereira una ciudad con tecnología de punta en el control y gestión del problema de movilidad, como también facilitar la monitorización y vigilancia del sistema vial en tiempo real, ayudando a disminuir los problemas de orden y seguridad, haciendo de Pereira una ciudad en la que de gusto transitar.

La metodología que se utilizó para desarrollar el proyecto, está enmarcada en la aplicación de procesos iterativos basados en un documento de requerimientos del sistema establecido en un acuerdo entre operarios del Observatorio de Movilidad Vial y el ingeniero de sistemas encargado del desarrollo. Luego en una etapa posterior el proceso continúa con un análisis de dichos

JORGE A. AGUIRRE

Ingeniero de Sistemas y Computación
Investigador
Universidad Tecnológica de Pereira
jorge@sirius.utp.edu.co

JOSE A. JARAMILLO

Ingeniero Electrónico, M Sc.
Docente de Facultad de Ingenierías
Universidad Tecnológica de Pereira
jj@sirius.utp.edu.co

requerimientos, un diseño del sistema y finalmente la implementación en un lenguaje de programación.

2. REQUERIMIENTOS DEL SISTEMA

Según la repartición funcional del Observatorio de Movilidad Vial de la ciudad de Pereira, el sistema de visualización es el encargado de mostrar, de la mejor manera posible, en una pantalla o series de pantallas, toda la información proveniente del sistema de información geográfico enfatizando en aquella información que tiene una alta frecuencia de actualización y necesita ser mostrada en tiempo real, con el fin de realizar un óptimo monitoreo del sistema vial de la ciudad de Pereira. Los requerimientos funcionales del sistema de visualización son los siguientes.

2.1 Descripción Funcional

El sistema de visualización del Observatorio de Movilidad Vial deberá tener tres funciones básicas particionadas en el siguiente diagrama:



Figura 1. Partición Funcional del Sistema

Visualización de Información Estática:

Hace referencia a la capacidad del sistema de visualización en poder mostrar al usuario final aquella información de la base de datos que posee una baja frecuencia de actualización o que por su relevancia para la gestión o naturaleza de poco cambio, permanece estática o almacenada por largos periodos de tiempo.

Visualización de Información Dinámica:

La visualización de información dinámica, hace referencia a la capacidad que tiene el sistema de mostrar datos externos en tiempo real. Los datos más relevantes provenientes del tráfico, se guardan en la base de datos del sistema de información geografía para pasar a estar disponibles para la consulta, análisis y estadísticas.

Generador de Estadísticas:

De acuerdo a los históricos de los datos alfanuméricos almacenados en la base de datos, se tiene la función de construir gráficas estadísticas a partir de consultas en

donde se estipule una fecha específica o un intervalo de fechas, como parámetro de entrada del buscador.

2.2 Representación de la Información

Definición de las Variables del Tráfico

Las variables del tráfico más relevantes que se van a monitorear, están conformadas por el conjunto de toda la información que proviene del sistema vial de la ciudad de Pereira y está inmersa dinámica o estáticamente en el sistema de información geográfica, estas variables son: Densidad vehicular, flujo vehicular, velocidad promedio, tendencia direccional de los vehículos, estado actual de los semáforos, programación actual de los semáforos.

Diagrama Entidad-Relación

Éste diagrama muestra la representación de la información y el cómo está conformado la estructura de las tablas de la base de datos. La información está clasificada de acuerdo a su estructura, distinguiendo los objetos geográficos de los objetos alfanuméricos, sin olvidar la relación entre ellos. El diagrama diseñado para el proyecto "Observatorio de Movilidad Vial".

Clasificación

El sistema de visualización tendrá en cuenta el tipo de información que será suministrada por el sistema de información geográfico para desplegarla de una manera óptima al usuario final, por esta razón se clasifica como estática la información que tiene una frecuencia de actualización baja con respecto al tiempo; dinámica la información que tiene una frecuencia de actualización que necesita ser visualizada en tiempo real y alfanumérica o geográfica si la información tiene o no tiene un componente geo-referenciado en el mundo real.

2.3 Requerimientos de Software

Motor de Base de Datos

El sistema de información geográfica que contiene todos los datos a mostrar en el sistema de visualización, está almacenado en el motor de bases de datos Oracle.

Sistema Operativo

El servidor en donde irá alojado el sistema de visualización con el sistema de información geográfico, estará sobre una plataforma Linux, de tal forma la implementación de la aplicación del sistema de visualización deberá ser compatible con dicha plataforma.

2.4 Requerimientos no Funcionales

A diferencia de los requerimientos funcionales directamente relacionados con el comportamiento específico del sistema, existen requerimientos no funcionales que buscan contemplar otras características que de una u otra forma pueden limitar y juzgar la operación sistema, estos requerimientos son:

Usabilidad

La aplicación debe diseñarse desde el punto de vista del usuario final, teniendo en cuenta que el objetivo del sistema de visualización es servir de herramienta para la gestión y administración. Por esta razón, se busca que todos los elementos del SIG tanto geográficos como alfanuméricos, tengan todas las herramientas necesarias para su fácil y correcta visualización.

Escalabilidad

El sistema de información geográfica del observatorio de movilidad vial, será un sistema de constante crecimiento durante la etapa de desarrollo del observatorio, por lo tanto el sistema de visualización, estará en las condiciones de mostrar toda la información nueva del SIG, en la medida que se esté actualizando.

Eficiencia

Debido a que el volumen de la información del tráfico, crecerá directamente proporcional al número de sistemas de medición ubicados en la malla vial por unidad de tiempo, el sistema de visualización estará en la capacidad de mostrar toda ésta información en tiempo real al usuario, sin ninguna disminución del rendimiento.

Confiabilidad

El sistema de visualización es un sistema que mantendrá en funcionamiento toda una jornada laboral, por lo tanto debe estar exento de errores que puedan estropear una correcta monitorización del tráfico de la ciudad.

3. ANALISIS Y DISEÑO

Para realizar el análisis de los requerimientos anteriormente presentados y el diseño del sistema, se escogió el UML, por ser el lenguaje más estándar y simple para modelar software orientado a objetos. Por esta razón se decidió analizar y diseñar el proyecto utilizando cada uno de los diagramas descritos en el lenguaje, abarcando así desde la especificación del sistema, hasta la formulación de un prototipo funcional.

En adición a UML, se implementó la metodología RUP (Proceso Unificado de Rational)¹, como la más estándar para el análisis, implementación y documentación de sistemas orientados a objetos. La aplicación de la

metodología RUP brindó al desarrollo del proyecto una mejor implementación de prácticas del software, una óptima administración de requisitos y un proceso de desarrollo iterativo, que ayudo al natural curso de la planificación y ejecución del proyecto.

Los diagramas utilizados descritos en [1], que permitieron desarrollar un modelo que sirviera como base para la implementación fueron los siguientes.

3.1 Diagramas de estructura

Diagrama de clases

En el diagrama de clases se ve reflejada la estructura del sistema, mostrando las clases, atributos y relaciones entre ellos, es un diseño conceptual de la información que se manejará en el sistema.

Diagrama de componentes

Este diagrama representa el cómo un sistema de software es dividido en componentes y muestra las dependencias entre ellos. Los componentes físicos de un sistema incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables y paquetes. Éstos diagramas prevalecen en el campo de la arquitectura de software, pero también pueden ser usados para modelar y documentar cualquier arquitectura del sistema.

Diagrama de despliegue

El diagrama de despliegue se utiliza para modelar el hardware de la implementación del sistema, como también las relaciones de cada uno de sus componentes.

Diagrama de paquetes

El diagrama de paquetes es utilizado para agrupar las clases a un nivel más alto, describiendo las dependencias entre cada uno de ellos.

3.2 Diagramas de Comportamiento

Diagrama de actividades

Un diagrama de actividades representa los flujos de trabajo paso a paso del negocio y las operaciones entre los componentes en un sistema.

Diagrama de casos de uso

Los diagramas de casos de uso, son una representación gráfica de todas y cada una de las interacciones típicas entre un actor y un sistema.

1. El **Proceso Unificado Racional** (*Rational Unified Process* en inglés, habitualmente resumido como **RUP**) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Diagrama de estados

Los diagramas de estados se utilizan para identificar cada una de las rutas o caminos que puede tomar un flujo de información después de ejecutarse un proceso. El diagrama de estados muestra en secuencia la ejecución de cada uno de los procesos del sistema

3.3 Diagramas de Interacción

Diagrama de secuencia

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase.

Diagrama de colaboración

El diagrama de colaboración tiene la función de visualizar la forma en que los objetos del sistema a desarrollar interactúan para alcanzar un objetivo, como también muestra las asociaciones o enlaces que tienen los objetos del sistema.

4. IMPLEMENTACION

4.1 Convención de Código

La implementación del diseño del sistema de visualización, se realizó en el lenguaje de programación Java, conservando los estándares que la red de desarrolladores de Sun propone a seguir. Los estándares están especificados en el documento Code Conventions for the Java Programming Language o Convención para la Programación en el Lenguaje Java [2], el cual contiene las convenciones de los nombres de archivos, organización de archivos, indentación, comentarios, declaraciones, espacios en blanco, convención en el nombramiento y buenas prácticas de programación.

4.2 Herramientas de Programación

NetBeans



Figura 2. Logo de NetBeans

Es un entorno integrado para el desarrollo de software (IDE), de código abierto y ofrece todas las herramientas necesarias para crear aplicaciones de escritorio, empresariales, web y móviles de manera profesional, utilizando el lenguaje Java, como también C/C++, PHP, Java Script, Groovy y Ruby.

NetBeans es así el entorno de desarrollo más popular, dada su simpleza en la instalación, su facilidad de manejo y la capacidad de integrarse a otras herramientas orientadas a la ingeniería del software.

ArcObjects

ArcObjects es un conjunto de librerías basadas en el modelo de datos geográficos escritas en C++. Ésta ofrece las herramientas necesarias para apoyar y construir soluciones SIG y son la base para que la familia de aplicaciones ArcGIS, se puedan adaptar a los requerimientos de los usuarios y desarrolladores de SIG. Los servicios del ArcObjects se clasifican en servicios base, acceso a datos, análisis de datos, presentación de mapas, componentes del desarrollador, marco para el desarrollo web, interfaz de usuario y extensiones; los cuales están organizados bajo las aplicaciones SIG, ArcGIS Desktop, ArcGIS Engine y ArcGIS Server, para adaptarlos a las necesidades de los desarrolladores SIG. En la siguiente ilustración se muestra la organización de los servicios según el producto:

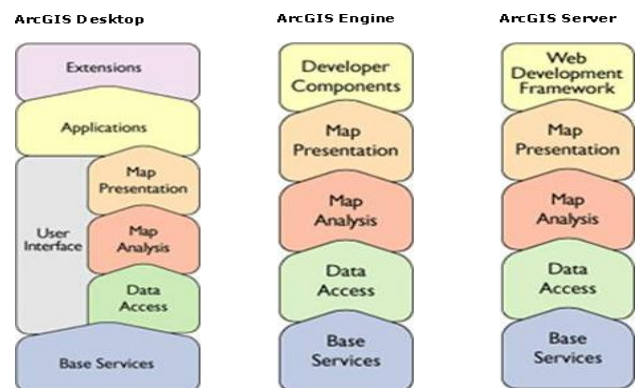


Figura 2. Distribución ArcGIS

5. CASOS DE PRUEBA

El objetivo de los casos de prueba dentro de un proyecto, es evaluar bajo diferentes contextos de ejecución, la funcionalidad de cada uno de los componentes de un sistema como el de su conjunto, informando al equipo desarrollador sobre las deficiencias, errores e inconformidades dentro del desarrollo. Para evaluar la funcionalidad de la aplicación de visualización, se escogieron cuatro tipos de pruebas (Operación, Negatividad, Basada en Requisitos, Rendimiento) establecidas en el libro [3], en donde cada una evaluará una dimensión diferente dentro del sistema desarrollado.

Las pruebas más relevantes para la comprobación de la funcionalidad del sistema desarrollado son:

5.1 Prueba de Operación del Sistema de Visualización

Esta clase de prueba se realiza para verificar el funcionamiento del sistema en condiciones normales en un largo periodo de tiempo, midiendo así la confiabilidad de la aplicación.

Caso de Prueba Único: Operación del Sistema de Visualización	
Propósito	Verificar que el sistema de visualización opere por un periodo de tiempo de un día laboral de trabajo bajo condiciones normales.
Prerrequisitos	Tener todos los sistemas del observatorio operando bajo circunstancias óptimas.
Pasos	Utilizar el sistema de visualización bajo condiciones normales de uso.
Resultados Esperados	La aplicación no modificará su comportamiento durante el tiempo transcurrido de la prueba.
Resultados Reales	La aplicación sostuvo el ritmo de trabajo de los operarios durante el periodo de tiempo de puesto en marcha, sin modificar su comportamiento.
Observaciones	Ninguna

Tabla 1. Caso de Prueba de Operación del Sistema de Visualización.

5.2 Prueba del Requisito de Tiempo Real

Esta prueba se realizó para comprobar que la aplicación desarrollada cumple con las especificaciones ofrecidas por el cliente.

Caso de Prueba por Requisitos: Visualización en Tiempo Real	
Propósito	Comprobar el funcionamiento del modo Tiempo Real, comparando los datos visualizados en tiempo real con los datos reales.
Prerrequisitos	Tener todos los sistemas del Observatorio de Movilidad Vial funcionando óptimamente.
Pasos	<ul style="list-style-type: none"> ✓ Ubicarse con una instancia de la aplicación en una intersección de la ciudad.
	<ul style="list-style-type: none"> ✓ Activar las capas de semáforos e intersecciones. ✓ Seleccionar en la aplicación la intersección visitada. ✓ Activar la opción de Tiempo Real. ✓ Realizar una comparación de la visualización de los cambios de los semáforos en

	la aplicación, con los cambios reales en la intersección. ✓ Medir los tiempos de retardo, si estos se presentan.
Resultados Esperados	Encontrar un retardo de menos de 2 segundos entre la visualización de la aplicación y los sucesos del mundo real. Tiempo que describirá el retardo de la transmisión de la información por el sistema de comunicación entre los sistemas de medición de variables del tráfico y el servidor de base de datos, y entre el servidor de la base de datos y la aplicación de visualización.
Resultados Reales	La diferencia entre los cambios reales de los semáforos y los cambios visualizados en la aplicación fue de 1.125 segundos.
Observaciones	Ninguno.

Tabla 2. Caso de Prueba por Requisitos de Visualización en Tiempo Real.

5.3 Prueba de Rendimiento de Ejecución sobre Diferentes Sistemas de Computo

Las pruebas de rendimiento o Pruebas de Capacidad tienen el objetivo de medir la capacidad de procesamiento del sistema bajo diferentes cargas de trabajo, incluyendo espacio de almacenamiento, uso de la CPU, etc. Los valores medidos se comparan con los valores requeridos.

Caso de Prueba de Rendimiento: Ejecución sobre Diferentes Sistemas de Computo.	
Propósito	Comprobar el funcionamiento de la aplicación en equipos de cómputo con características diferentes.
Prerrequisitos	Tener toda la plataforma de ejecución configurada sobre los equipos de cómputo de prueba.
Pasos	Ejecutar la aplicación en los diferentes equipos de cómputo en instantes diferentes. Evaluar el desempeño del modo de tiempo real y la velocidad de renderizado de la capa Semáforos y el Mapa Base.
Resultados Esperados	Entre más potente sea el sistema de cómputo, el modo de tiempo real será más fiel a los cambios de la base de datos y la velocidad de renderizado de las capas geográficas será más alta.
Resultados Reales	El sistema solo se ha probado en los siguientes equipos: <ul style="list-style-type: none"> • HP Procesador: Pentium 4 RAM: 1 Gb, Disco Duro: 80

1. El **Proceso Unificado Racional** (*Rational Unified Process* en inglés, habitualmente resumido como **RUP**) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

	<p>Gb</p> <ul style="list-style-type: none"> • Dell Inspiron Procesador Intel i5 2.13 Ghz RAM 4 Gb Disco Duro 320 Gb Tarjeta Grafica: ATI Radeon 4300 Series de 512 Mb. • Dell Studio Procesador Core 2 Duo 2.4 Ghz RAM 3 Gb Disco Duro 320 Gb Tarjeta Gráfica: ATI Radeon 4500 Series 256 Mb. <p>Los resultados en cuanto a la velocidad de actualización del modo en tiempo real, fueron despreciables entre los tres equipos.</p> <p>Los resultados del tiempo de carga y renderizado del Mapa base fueron los siguientes:</p> <ul style="list-style-type: none"> • HP: 15 Segundos • Dell Inspiron: 5 Segundos • Dell Studio: 7 Segundos <p>El rendimiento de la aplicación en cuanto a la carga de elementos geográficos varía de acuerdo a la potencia del procesador del equipo como en la capacidad de su tarjeta gráfica. Entre más potente el equipo más rápido realiza la carga de los elementos.</p>
Observaciones	<p>El rendimiento de la aplicación en cuanto a la carga de elementos geográficos varía de acuerdo a la potencia del procesador del equipo como en la capacidad de su tarjeta gráfica. Entre más potente el equipo más rápido realiza la carga de los elementos.</p> <p>La aplicación no se ha podido ejecutar en el equipo de visualización del Observatorio de Movilidad Vial, debido a la disponibilidad de este.</p>

Tabla 3. Caso de Prueba de Rendimiento de Ejecución sobre Diferentes Sistemas de Computo

tanto la implantación de la aplicación desarrollada debe hacerse sobre una plataforma Windows.

- ✓ Dadas las características de la información en tiempo real y las dificultades en el desarrollo que en su entorno contiene, el prototipo desarrollado enfrenta el problema con éxito mostrando solo un retardo de 1 segundo desde el momento que los sucesos reales ocurren, hasta que son visualizados.
- ✓ Después de numerosas pruebas, el prototipo desarrollado cumple con todos los requerimientos del Observatorio de Movilidad Vial anunciados en el inicio del proyecto, con futuras propuestas encaminadas a la formación del producto final.

7. BIBLIOGRAFÍA

- [1] M. Fowler y K. Scott, *UML gota a gota*, México: Addison Wesley Longman de México, S.a., 1999.
- [2] Sun Microsystems, Inc. (1997, Sept.). Code Conventions for the Java Programming Language, Sun Microsystems, Inc. [Online]. Available: <http://java.sun.com/docs/codeconv/>
- [3] A. Weitzenfeld, *Ingeniería de software orientada a objetos con uml, java e internet*, Ed. México: International Thomson Editores, S.A., 2005.
- [4] ESRI. (2007, Mar). ArcGIS 9.2 Desktop Help. [Online]. Available <http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=welcome>
- [5] ESRI. (2004). ArcGIS Engine Developer Guide. Redlands CA.
- [6] E. Hull, K. Jackson y J. Dick, *Requirements*, 2nd Ed, USA : Springer Science+Business Media, 2005.
- [7] R. Young, *The Requirements Engineering Handbook*, Norwood, MA: Artech House, Inc., 2004.
- [10] R. Pressman, *Ingeniería del Software: Un Enfoque Práctico*. México: McGraw-Hill Interamericana Editores, S.A., 2005.

6. CONCLUSIONES Y RECOMENDACIONES

- ✓ Para realizar una óptima visualización de los elementos del sistema vial, se debe tener una plataforma SIG, en donde se encuentre toda la información geográfica y alfanumérica de los elementos que se quieren visualizar. Dicha plataforma brindará al usuario final el principal elemento gráfico para realizar el monitoreo de las variables del tráfico de la ciudad.
- ✓ Existe incompatibilidad de los componentes del ArcGIS en Java con el sistema operativo Linux, por